Vol.2015-CE-128 No.12
2015/2/14

タブレットとスマートウォッチに対応した数値計算ツールを用いた制御工学教育支援

川端 悠一郎[†1]　古賀 雅伸[†1]　津村 祐司[†1]　矢野 健太郎[†2]

概要：本研究では，制御工学において多用される行列などの数式を効率的に扱える数値計算言語 MATX を Android 搭載のタブレットや Android Wear 搭載のスマートウォッチで利用できる数値計算ツール MATX mobile を開発した．本ツールを使用することにより制御工学教育の演習や実験における制約が緩和されるだけでなく，教育効率の向上が期待できる．

1. はじめに

近年，タブレットなどのスマートデバイスが普及しており，スマートデバイスで業務をこなすことができる実用的なアプリケーションや e-ラーニングといった教育用のアプリケーションの開発が盛んに行われている．

数値計算プログラムや制御系設計においてもスマートデバイスで開発が行えるような環境が徐々に整ってきている．スマートデバイスは小型軽量で携帯しやすいので, 数値計算や制御系設計のための計算を行える時や場所の制約が大幅に緩和される．

スマートデバイスでは現在 MATLAB mobile[1], Maxima on Android[2], Octave[3], addi[4] といった数値計算ツールがリリースされている．

MATLAB mobile はスマートデバイス上で MATLAB[5] を扱うことのできるアプリケーションである．提供しているのは，通信するための手段と計算を入力・表示するための UI である．実際の計算は自身の PC，または MathWorks Cloud で実行される．

Maxima on Android は数値計算ライブラリ Maxima を Android デバイスで利用可能にしたアプリケーションである．UI としてコンソールが提供されている．

Octave は数値計算ライブラリである Octave[6] を Android デバイスから利用可能にした数値計算ツールである．UI としてコンソールが提供されている．

addi は Android で利用できる MATLAB クローンである．addi の処理も同様に Octave という数値計算ライブラリを使用している．UI としてコンソールが提供されている．

これらの数値計算ツールは行列等の複雑な計算や変数を使用することが可能であり，複雑な計算を要する制御系設計等にも応用可能である．

本研究では, 数学的表現に近い形でプログラムを記述できる数値計算言語 MATX[7] を Android 搭載のスマートデバイスでの利用できる数値計算ツール MATX mobile を提案し，実装する．そして，数値計算プログラムを用いての制御系シミュレーションを可能とし，制御教育の現場への導入を目指す．

一方スマートデバイスにはハードウェアキーボードがなく PC に比べで入力効率が悪いという問題が指摘されているが, 本研究では数値計算の入力に特化した独自のソフトウェアキーボードやタッチパネルを利用した補完機能などの UI を開発することでこの問題を解決する．

また，本研究では近年普及しつつあるウェアラブルデバイスとの連携を可能とすることで，スマートデバイスでの数値計算ツールの利用の幅を広げる．

2. スマートデバイスにおける制御系設計と教育現場への導入

本研究で提案する数値計算ツールは制御系設計に応用可能であり，制御教育の現場に導入することで教育の質を向上させられると考える．

導入は九州工業大学　情報工学部　システム創成情報工学科　3 年次に開講される実験を想定している．

2.1 実験内容

実験は 22 名で開講され，4〜5 名 1 グループに分かれて

[1] 情報処理学会
IPSJ, Chiyoda, Tokyo 101–0062, Japan
[†1] 現在，九州工業大学
[†2] 現在，福岡工業大学短期大学部

制御実験を行う．

実験では制御対象として，図 1 に示されるような倒立振子系を考える．モノレール上に台車が置かれ，台車上のモノレールと直角な軸に 1 本の棒が取り付けられ，棒はその軸まわりに自由に回転できる．台車はベルトとプーリを介して，モータにより駆動され，モノレール上を走行できる．すなわち，棒 (振子) は鉛直線とモノレールにより定まる平面に拘束されて，台車によって動かされるようになっている．

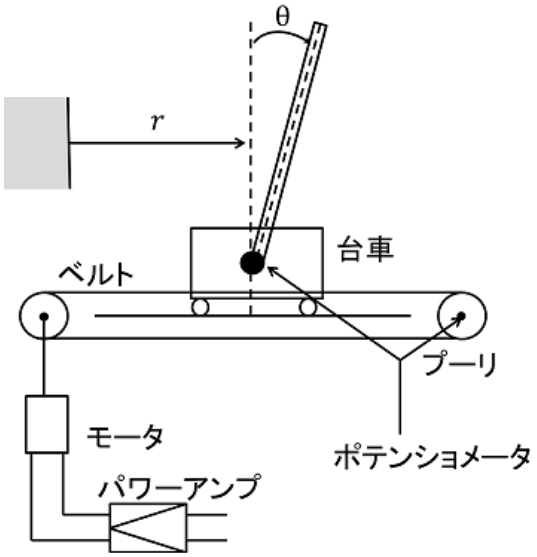

図 1 倒立振子制御系の実験装置

また，実際の実験装置の写真を図 2 に示す．

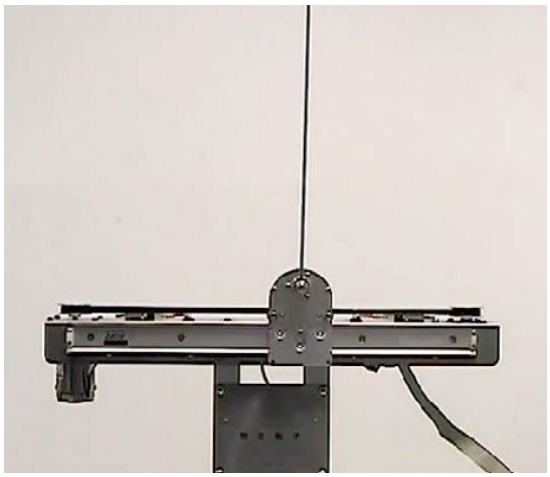

図 2 倒立振子制御系の実験装置 (写真)

2.2 制御目的

この倒立振子系は，2 つの平衡点もつ．1 つは棒が鉛直線に沿って垂れ下がった状態，もう 1 つは棒が鉛直線に沿って倒立した状態である．前者は，棒を揺らせれば，いわゆる振子となり，揺れは時間が立てば止まるので，安定平衡点である．後者は，いわゆる倒立振子であるが，倒立状態にある棒を少しでもつつけば，真っ逆さまに落ちて行くので，不安定平衡点である。このような倒立振子系に対する制御目的として，つぎを考える．

- 倒立状態にある棒が何らかの原因で傾いたとき，台車を動かして，棒をすみやかに倒立状態に戻す (不安定平衡点の安定化)．
- 倒立位置を指定した位置に移動させる．

2.3 実験プロセス

この実験は制御対象の物理パラメータを測定し，制御対象が制御可能かを解析する．解析後は設計を行いそれに基づきシミュレーション計算を行う．

もし，シミュレーションが期待通りの結果出なかった場合はもう一度設計からやり直す．

シミュレーションの結果が期待通りであれば実機を制御し，失敗したなら再度設計を行うというプロセスがある．

実験のプロセスを図 3 に示す．

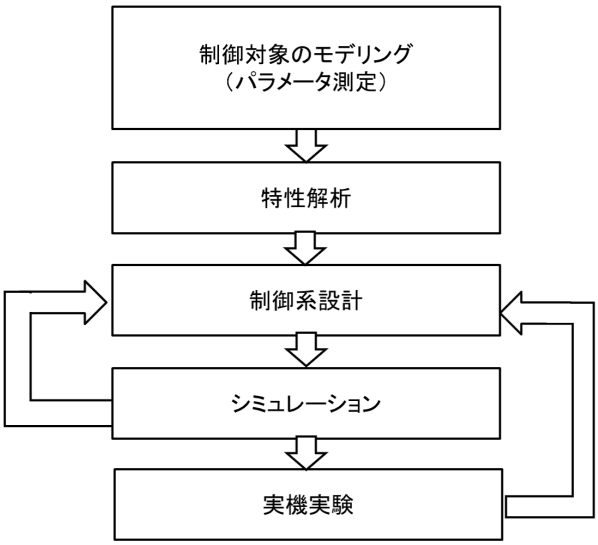

図 3 実験プロセス

2.4 課題

実験は 4〜5 人グループで作業を行うが，全員に作業を分担することが難しく，作業をしていない学生が存在してしまうことが課題として挙げられる．

図 4 は授業風景の中で発生している課題である．

この図では右上の生徒や左の生徒は PC での実験にうまく参加でいない．このようにすべての学生が実験に参加できているといえない現状がある．

2.5 スマートデバイス導入の利点

2.5.1 利用場所の制約の緩和

スマートデバイスは小型で軽量であり，利用場所の制約

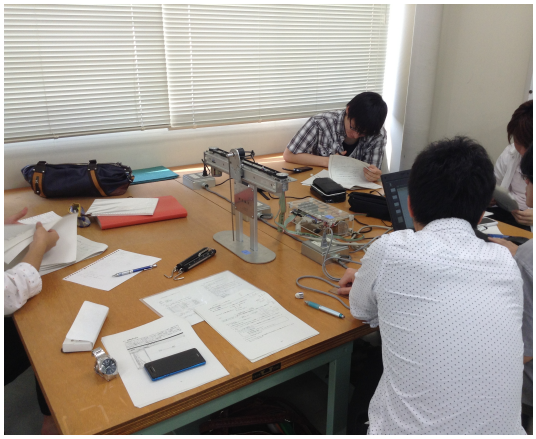

図 4 プロジェクト風景

が緩和される．例えば机のない環境でも立ったまま利用することができる．

また，計算処理にネットワーク環境が必要な数値計算ツールの場合は場所の制約が少しかかってしまうが，本研究で実装するツールは計算処理をツール内部に記述しているためネットワーク環境は必須ではない．

2.5.2 操作性

計算した結果をグラフに表示する際，スマートデバイスでは拡大や縮小の操作がタッチ操作で直感的に行える．

2.5.3 所有率の高さ

近年ではタブレットやスマートフォンの所有率が毎年増加しており，人々にとって馴染みの深いデバイスとなっている．

大学生については，2014 年は 98 %がスマートフォンを所有しているというデータがある．

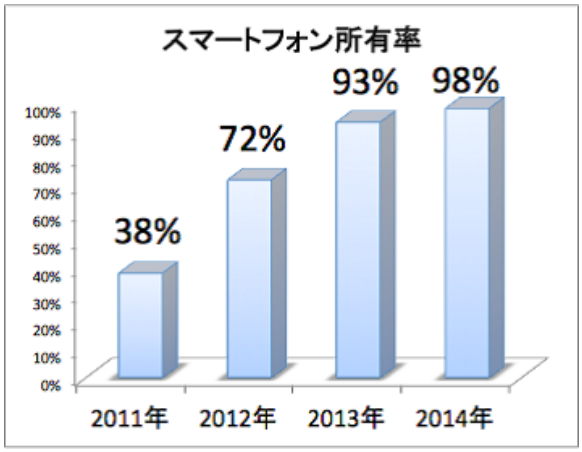

図 5 スマートフォン普及率

加えて近年の学生は PC とスマートデバイスで同様のことができるならば，スマートデバイスで行う方を好む傾向にある．[9] このことからスマートデバイスを用いて制御系設計を行える数値計算ツールを学生に配布し教育に導入することで，学生は簡単に授業の課題や実験のシミュレーション等を行う事ができるようになると考える．

2.5.4 課題の解決

導入を想定している実験の課題として，学生全員に作業を割り振れないというものがある．その課題は主に解析やシミュレーション等 PC を使用した作業を行う際に発生する．

そこで，この作業をを PC だけではなくスマートデバイスで行えるようにすることで，すべての学生に作業を分担することができる．

スマートデバイスを導入する実験のプロセスを図 6 に示す．

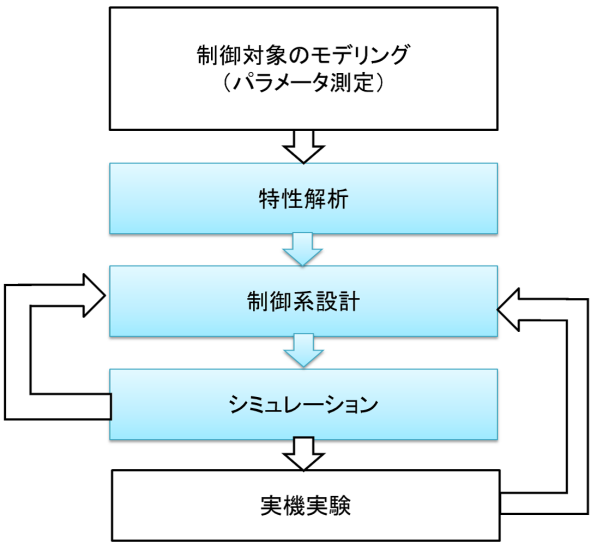

図 6 スマートデバイスを導入するプロセス

このプロセスにスマートデバイスを導入することで図 7 のように学生全体が作業を行うことができるようになると期待できる．

図 7 実験にスマートデバイス導入後のイメージ

また，学生の所有するスマートデバイスで動作することで，これまで行えなかった予習・復習を行うことができるため，効率よく課題に取り掛かることができ，実験の質も向上すると考える．

2.6 スマートデバイス導入における課題

2.6.1 入力速度

スマートデバイスで数値計算を行う際の入力は PC とは異なり，ソフトウェアキーボードを用いて入力することが多い．

しかし，標準搭載されているソフトウェアキーボードでは数値計算を行う際に頻出する算術記号や括弧などの入力に非常に手間がかかってしまい著しく入力速度が低下してしまう．

通常搭載されているソフトウェアキーボードの例を図 8 に示す．

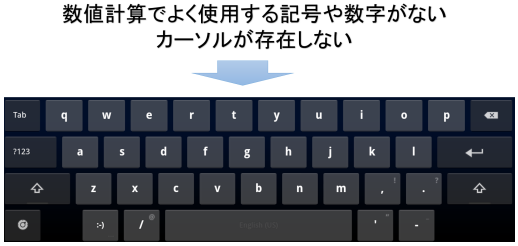

図 8 標準搭載されいるソフトウェアキーボードの例

2.6.2 計算速度

スマートデバイスで数値計算を行う際は計算速度は PC と比べると速度が劣ってしまう．

スマートデバイスでの数値計算で時間を要してしまった場合，スマートデバイスを常に確認していなければならず，時間的拘束を受けてしまうという課題がある．

3. スマートデバイスにおける制御系設計に応用可能な数値計算ツールの実装

本研究では制御系設計に応用できる数値計算ツール MATX mobile を実装した．

実装した MATX mobile の全体像を図 9 に示す．

図 9 MATX mobile の全体像

MATX mobile は計算結果を入力・表示するコンソール部分を左上に，ジェスチャー操作を検知するパッド部分を右上に，キーボードを下に配置している．

本章では実装したツールについて詳しく述べる．

3.1 数値計算エンジン JMATX の利用

スマートデバイスで数値計算ツールを扱う際 UI とエンジンが独立している場合が多い．

本研究で提案する数値計算ツールは独自 UI を開発し，UI からエンジンを利用する構造とする．本研究で提案する数値計算ツール MATX mobile の構造イメージを図 10 に示す．

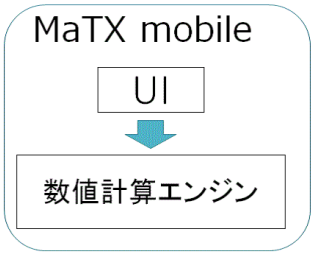

図 10 MATX mobile の構造イメージ

本研究で実装した数値計算ツールでは，数値計算エンジン JMATX[8] を利用する．JMATX はすべて java 言語で実装されており，Android OS 上のバーチャルマシンでも，PC での利用と同様な挙動を可能としている．

3.2 ソフトウェアキーボードの実装

本研究では通常のキーボードよりも入力効率を上げるためにソフトウェアキーボードを実装する．

キーボードのレイアウト

スマートデバイスで数値計算を扱う際には，細かい計算式の編集が頻発することが予測されるため，細かい編集を容易に行えるカーソルを右下に配置する．また，数値計算を行う際に入力頻度が高くなる算術演算子や括弧などはキーボードをなるべく切り替えずに入力できるようにキーの上部に配置する．

カーソルの自動移動

本研究で提案するキーボードには ”()” ， ”[]” のような両括弧を入力できるキーを配置する．これらのキーをタップすることで両括弧が入力され，カーソルは括弧の中に自動的に移動できる機能を提案する．

これにより，数値計算言語で起こってしまう括弧の不整合を防ぐことができると考える．

実装したソフトウェアキーボード

実装した UI から入力された計算式を JMATX を利用して計算し，その結果をコンソール部に表示する．

ソフトウェアキーボードは java 言語と xml ファイルで構築し，数値計算ツールとは独立させて実装した．そのた

め容易に他の数値計算ツールや制御系 CAD に適応することができる．

実装したキーボードを図 11 に示す．

図 11 MATX mobile キーボード

テンキーボード

上記のキーボードと同様にテンキーボードを実装した．実装したキーボードを図 12 に示す．

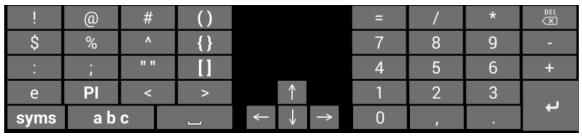

図 12 テンキーボード

このテンキーボードは標準のキーボードから切り替えることで表示ができる．

3.3 入力補完機能の実装

2 章で述べた課題である入力速度を解決するために入力効率を向上させる UI を提案する．

MATX には多くの内部関数が用意されている．しかし，ソフトウェアキーボードでは長い名前の関数や，一度使用した名前の長い変数などをタイプするのに PC で扱うハードウェアキーボードでの入力に比べて効率が遅くなってしまう．その対策として入力途中の関数名や変数名を補完できる入力補完機能を提案する．

例えば，【isempty】という関数を利用したい場合は　is　まで入力が完了した時点で is から始まる関数を列挙し，タップ操作により選択可能にするビューを作成する．

そして，タップすることにより【isempty()】を自動的に入力しカーソルを自動的に括弧内に移動させることで関数の入力を効率よく行えると考える．

3.3.1 実装

入力補完機能はコンソールの入力部の上部に表示するように実装した．操作画面を図 13 に示す．

図 13 入力補完

3.4 行列入力エディタ

本研究で扱う数値計算言語 MATX では行列の入力は以下のように表わすことができる．

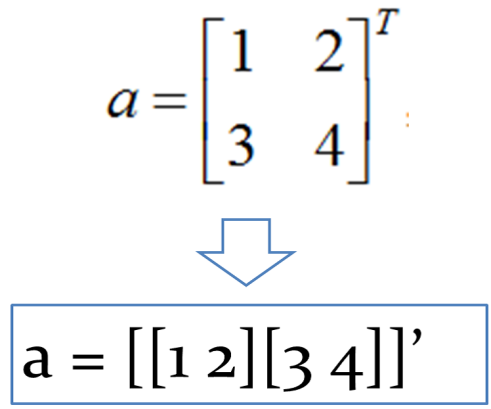

図 14 MATX での行列の表現方法

行列をソフトウェアキーボードで入力しようと思うと成分の数が増すにつれタイプミスする可能性がある．その対策として，行列用の入力エディタを提案する．

エディタは行列の要素数を入力し，その後行列の要素を入力することで文字列を生成し，入力部に自動的に入力される．

これにより行列の括弧による不整合や要素の入力ミスを防ぐことができると考える．

3.4.1 実装

行列入力エディタ MATX mobile の右部分に, 使用する際のみ表示するように実装した．エディタに入力が完了すればコンソールの入力部に行列式が自動的に入力される．

操作画面を図 15 に示す．

図 15 行列入力ダイアログ

3.5 ジェスチャー操作機能

通常コンソールを PC で扱う場合はカーソルの上矢印キーを押すことで入力履歴が閲覧できる場合が多い．

しかし，ソフトウェアキーボードに搭載されているカーソルの上矢印キーはハードウェアキーボードに比べて小さくタッチミスが多くなってしまう．そこで，スマートデバイス上ではジェスチャーを用いて入力履歴の閲覧等よく利用する動作をミスなく行える UI を提案する．

3.5.1 実装

タッチジェスチャーを検知するためのタッチ部分はツールの右側に配置した．右の白いパット部を指でなぞることでその動きを検知し，履歴の取得や文字の拡大・縮小を行えるよう実装した．

操作画面を図 16 に示す．

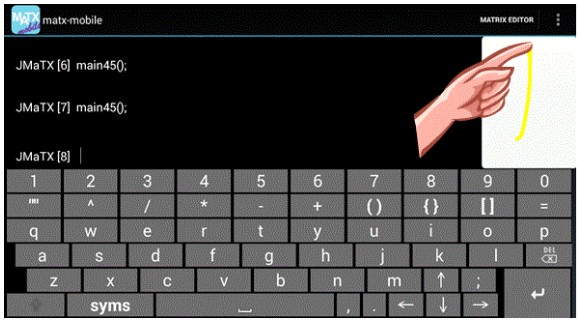

図 16 ジェスチャー機能

3.6 計算速度の課題の緩和

計算速度の課題の緩和のために，本研究ではウェアラブルデバイス（腕時計型）との連携を提案する．

利用想定を図 17 に示す．

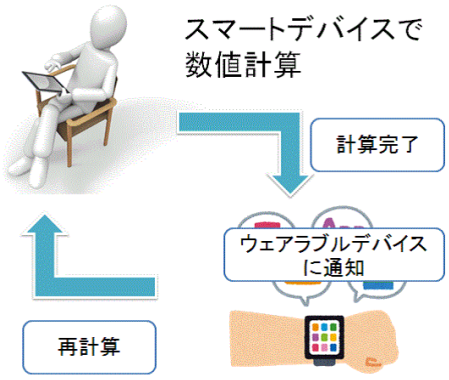

図 17 ウェアラブルデバイスとの提案

ウェアラブルデバイスと連携することで，計算時間を長く要した場合はスマートデバイスを常に監視していなくても計算結果を通知することができるので待機時間の負担を緩和できると考えている．

本研究では，今後普及していくと考えられるウェアラブルデバイスとの連携をすることで，ツールの利便性を向上させられると考えている．

3.7 繰り返し計算の実行

通知された計算結果画面から再び再計算要求を行える繰り返し計算機能をウェアラバるデバイスに発行する Notification に実装した．

関数として入力された結果に対しては関数の引数を変更して、ウェアラブルデバイスからスマート端末に計算要求っを行うことを可能とした．

流れを図 18 に示す．

図 18 繰り返し計算の流れ

また、その他の入力を行う際は Google が提供している音声入力機能を利用して入力が可能にした．

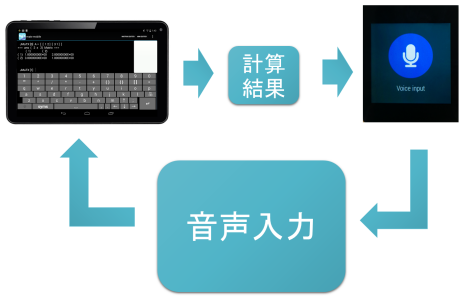

図 19 繰り返し計算の流れ（音声入力）

4. 評価

実装した数値計算ツールの制御教育の現場への導入検証，UI の評価，計算時間の評価を行う．

4.1 制御教育現場への導入

本稿のはじめに述べた制御教育の現場である実験への導入が可能か検証する．

検証として，実際の実験で制御対象とする倒立振子のシミュレーション計算を行う．

制御対象

制御対象として図 1 に示されるような倒立振子系を考える．台車を目標値に移動させながら，振子をすみやかに倒立状態に戻す．また，台車の目標値を 5 秒ごとにステップ状に変化する．

シミュレーション実行

シミュレーションは状態を計算する微分方程式，台車への出力を計算する出力方程式を用意し，ode45 ソルバーを用いて行う．微分方程式関数，出力方程式関数，シミュレーションを行うメイン関数を用意し，15 秒間のシミュレーションを行う．関数の作成には MATX mobile の UI を使用することができる．今回使用する関数の編集画面を図 20 に示す．

図 20 関数の編集

メイン関数を実行することで変数 T にシミュレーション時間，変数 X に状態，変数 U に台車への出力が格納される．

シミュレーション結果の可視化

本研究で開発した MATX mobile は外部アプリケーションと連携することでグラフの出力を可能としている．これを用いて例題で導出したシミュレーション結果を可視化する．

グラフを出力するために実装した plot 関数を用いる．plot 関数を以下に示す．

グラフ出力関数

```
plot(T,X(1,*));
```

台車の変位 r のシミュレーション結果を図 21 に示す．

この結果は PC で行ったシミュレーションと同様のものを得ることができた．

このことからスマートデバイスを用いて倒立振子のシミュレーションを行うことができ，制御教育の現場で本研究で開発したツールを導入することは可能という事が確認できた．

4.2 計算速度比較

上記の例題のシミュレーション時間を 15 秒に設定して，PC，スマートデバイスごとに計算時間の比較を行った．

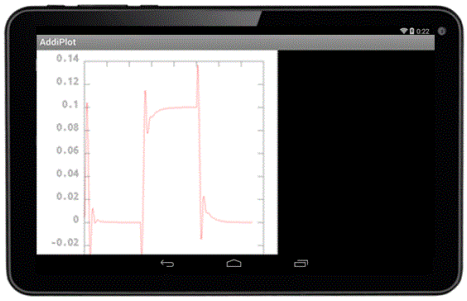

図 21 台車の変位 r のシミュレーション結果

今回は PC，スマートデバイスは 2011 年に発売された Android OS を搭載したタブレット，2013 年に発売された Android OS を搭載したタブレット端末， そして 2013 年発売のタブレットで Android OS で新たに提案されているランタイムである ART (Android Runtime) を使用して実行した場合を比較する．

ART (Android Runtime)

ART は AOT (Ahead of Time) コンパイル形式のランタイムである．

従来のランタイムはコードをコンパイルして Android アプリを動作させる Just in Time 方式を使用していたが，ART はアプリケーションのインストール時にあらかじめ DEX コードをコンパイルした機械語コードを保存しておき、アプリ起動時はこの機械語コードを実行する Ahead-Of-Time 方式を採用している．

これにより搭載された CPU の能力を効率よく利用することができる．

実行環境

実行環境を表 1 から表 3 に示す．

表 1 PC 実行環境

	環境
CPU	Intel(R) Core(TM) i5-4570S CPU @ 2.90GHz
メモリ	16.0GB
OS	Windows 8.1 Enterprise 64bit
Java VM	Java(TM) SE Runtime Environment (build 1.8.0_b132)

表 2 Android1(2011 発売) 実行環境

	環境
CPU	ITegra 2 1GHz
メモリ	1024 MB
OS	Android 3.2.1

表 3　Android2(2013 発売)　実行環境

	環境
CPU	Qualcomm Snapdragon? S4 Pro 1.5 GHz
メモリ	2048 MB
OS	Android 4.4.4

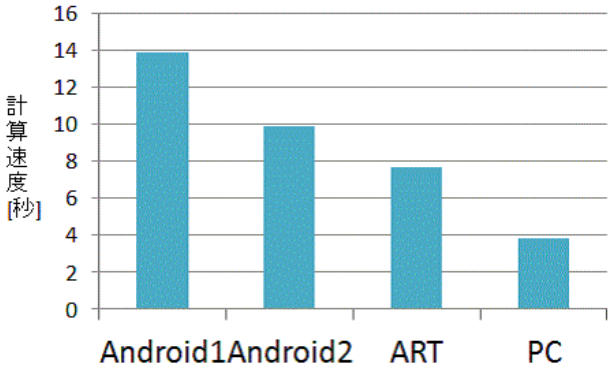

図 22　速度比較

4.2.1　実行時間比較

計算時間の比較を図 22 に示す.

現在計算速度は現在 PC と比べると差が発生してしまっているが，Android デバイスのスペックの向上により PC に追いつきつつあることが結果からわかる.

今回評価したランタイムである ART は今後発売される Android 端末で採用される予定があり，今後スマートデバイスの計算速度は PC と同様まで追いつくことが考えられる.

4.3　UI 評価

今回提案した UI による入力を行いその速度を測定することで評価を行う.

下記のような入力を MATX mobile に搭載されている Editor を使用し, 提案した UI を使用して入力した場合と使用せずに入力場合に要する時間を比較する.

計算プログラム

```
Func Matrix LQR(){
   q0 = [1e6 1e3 1 1];
   Q=vec2diag(q0);
   R = [1];
   A = [[0,0,1,0]
   [0,0,0,1]
   [0,-1.17e-1,-1.80e1,1.00e-3]
   [0,-4.76e-1,8.80e1,-4.11e-1]];
   B = [[0][0][1.27][-6.13]];
   {F,P}=lqr(A,B,Q,R);
}
```

上記の記述を学生 8 名に実装した UI を使用した場合と UI を使用しない場合で入力してもらった．平均入力時間，文字修正回数の平均は表 4 のようになった.

表 4　平均入力時間, 平均文字修正回数

	入力時間	文字修正回数
UI 不使用	296.8 秒	3.6 回
UI 使用	409.8 秒	3.0 回

修正回数には大きな向上は見られなかったが，実装した UI を使用することにより入力速度は向上することが確認できた.

5.　おわりに

本研究では制御教育現場へ導入するための数値計算ツールを開発し，ウェアラブルデバイスとの連携を可能にした．そして，導入検証をを行い，導入可能という事を確認した今後は，本ツールを制御教育現場へ導入し，アンケートなどの評価を行い，それに基づいた改良を行っていきたい.

開発したツールは Google Play に公開中で図 23 の QR-code に記載された URL からダウンロード可能にした.

図 23　MaTX-mobile ダウンロードページ

参考文献

[1] MathWorks. Matlab mobile home page. http://www.mathworks.co.jp/mobile/.
[2] YasuakiHonda. Maxima on android github. https://github.com/YasuakiHonda/Maxima-on-Android/.
[3] CORBIN CHAMPION. Octave4android github. https://github.com/corbinlc/octave4android/.
[4] CORBIN CHAMPION. addi home page. https://code.google.com/p/addi/.
[5] MathWorks. Mathworks home page. http://www.mathworks.co.jp/.
[6] John W. Eaton. Gnu octave home page. http://www.gnu.org/software/octave/.
[7] 古賀雅伸. Matx home page. http://www.matx.org/.
[8] 杉永良太. 精度保証付き数値計算に対応した数値計算言語の開発と制御系設計への応用. 九州工業大学, 2011.
[9] 坂本　毅啓, 佐藤　貴之. スマートフォンによるコミュニケーションスキル獲得を目指した教材の開発. 教育システム情報学会　第 4 回研究会. 2014. 49-54.